Claims

1. A computer implemented method for automatic software tuning comprising the steps of:
calculating (410) at least one threshold value for at least one parameter (P1) influencing the performance of a software application (200) with regards to a specific task;
comparing (430) the at least one threshold value to at least one corresponding current value; and
selecting (440) an algorithm (A1) from a plurality of algorithms (A1 to AN) for performing the task in accordance with the result of the comparing step (430).

2. Method of claim 1 comprising the further steps of:
measuring (450) the performance of the selected algorithm (A1);
checking (460) whether the measured performance complies with the at least one threshold value; and
recalculating (470) the at least one threshold value in case of non-compliance.

3. Method of any one of the previous claims, where the at least one threshold value separates the value range of the parameter (P1) into at least two intervals of a first dimension.

4. Method of claim 3, wherein the selecting step (440) selects the algorithm (A1) that is assigned to the interval that includes the corresponding current value of the first dimension.

5. Method of claim 3, where at least one further threshold value separates the value range of a further parameter into at least two intervals of a second dimension.

6. Method of claim 5, wherein the selecting step (440) selects the algorithm (A1) that is assigned to the intersection of the interval of the first dimension that includes the corresponding current parameter value of the first dimension and the interval of the second dimension that includes the corresponding current parameter value of the second dimension.

7. Method of any one of the claims 3 to 6, wherein each threshold value corresponds to a break-even point where two neighbouring algorithms have the same performance with respect to the corresponding dimension.

8. A computer program product for automatic software tuning comprising a plurality of instructions that when loaded into a memory of a computer system (990) cause at least one processor of the computer system (900) to execute the steps of any one of the claims 1 to 7.

9. Information carrier comprising the computer program product of claim 8.

10. A computer program product for dynamically selecting a data retriever implementation for retrieving data from a data storage system (902) in response to a Boolean expression (500) comprising:

a result counter (102) to determine a number of hits in response to the Boolean expression;

a threshold evaluator (103) to compare the number of hits with a threshold value of a first dimension and to compare the complexity of the Boolean expression with a further threshold value of a second dimension;

a first data retriever (111) to retrieve the data in case the number of hits is below the threshold value of the first dimension and the complexity of the Boolean expression is above the further threshold value of the second dimension;

a second data retriever (112) to retrieve the data in case the number of hits is above the threshold value of the first dimension and the complexity of the Boolean expression is above the further threshold value of the second dimension;

a third data retriever (113) to retrieve the data in case the number of hits is below the threshold value of the first dimension and the complexity of the Boolean expression is below the further threshold value of the second dimension; and

a forth data retriever (114) to retrieve the data in case the number of hits is above the threshold value of the first dimension and the complexity of the Boolean expression is below the further threshold value of the second dimension.

11. The computer program product of claim 10, further comprising:

a retrieval time measuring component (104) to measure the time that is consumed by a selected data retriever (111, 112, 113, 114) for various numbers of hits; and

a threshold calculator (105) to dynamically determine the threshold value and the further threshold value on the basis of the results of the retrieval time measuring component (104) and to feed back the determined threshold values into the threshold evaluator (103).

12. The computer program product according to claim 11, where the first data retriever (111) is implemented by using a general data retrieval algorithm using result flag instances.

13. The computer program product according to claim 11 or 12, where the second data retriever (112) is implemented by using a general data retrieval algorithm using bit maps.

14. The computer program product according to any one of the claims 11 to 13, where the third data retriever (113) is implemented by using a lean AND data retrieval algorithm using result flag instances.

15. The computer program product according to any one of the claims 11 to 14, where the forth data retriever (114) is implemented by using a lean AND data retrieval algorithm using bit maps.

16. A computer system (990) comprising:
    a memory to store a computer program product according to any one of the claims 10 to 15; and
    at least one processor to execute instructions of the computer program product according to any one of the claims 10 to 15.

17. A computer system (990) for running a software application (200) comprising:
    variables (210) for storing at least one threshold value for at least one parameter (P1) influencing the performance of the software application (200) with regards to a specific task; and
    a threshold evaluator (220) for comparing (430) the at least one threshold value to at least one corresponding current value allowing the software application (200) to select (440) an algorithm (A1) from a plurality of algorithms (A1 to AN) for performing the task in accordance with the result of comparison.

18. The computer system (990) of claim 17, further comprising:
    a threshold calculator (230) for recalculating (470) the at least one threshold value in case the actual performance of the selected algorithm (A1) is non-compliant with the at least one threshold value.

19. The computer system (990) of claim 17 or 18, where the at least one threshold value separates the value range of the parameter (P1) into at least two intervals of a first dimension.

20. The computer system (990) of claim 19, wherein the selected algorithm (A1) is assigned to the interval that includes the corresponding current value of the first dimension.

21. The computer system (990) of claim 19, where at least one further threshold value separates the value range of a further parameter into at least two intervals of a second dimension.

22. The computer system (990) of claim 21, wherein the selected algorithm (A1) is assigned to the intersection of the interval of the first dimension that that includes the corresponding current parameter value of the first dimension and the interval of the second dimension that that includes the corresponding current parameter value of the second dimension.

23. The computer system (990) of any one of the claims 19 to 22, wherein each threshold value corresponds to a break-even point where two neighbouring algorithms have the same performance with respect to the corresponding dimension.